

A cubic vertex kernel for Diamond-free Edge Deletion and more^{*}

R. B. Sandeep^{1**} and Naveen Sivadasan²

¹ Department of Computer Science & Engineering
Indian Institute of Technology Hyderabad, India
cs12p0001@iith.ac.in

² TCS Innovation Labs, Hyderabad, India
naveen@atc.tcs.com

Abstract. A diamond is a graph obtained by removing an edge from a complete graph on four vertices. A graph is diamond-free if it does not contain an induced diamond. The DIAMOND-FREE EDGE DELETION problem asks whether there exist at most k edges in the input graph G whose deletion results in a diamond-free graph. For this problem, a polynomial kernel of $O(k^4)$ vertices was found by Fellows et. al. (Discrete Optimization, 2011).

In this paper, we give an improved kernel of $O(k^3)$ vertices for DIAMOND-FREE EDGE DELETION. Further, we give an $O(k^2)$ vertex kernel for a related problem $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION, where $t \geq 4$ is any fixed integer. To complement our results, we prove that these problems are NP-complete even for K_4 -free graphs and can be solved neither in subexponential time (i.e., $2^{o(|G|)}$) nor in parameterized subexponential time (i.e., $2^{o(k)} \cdot |G|^{O(1)}$), unless Exponential Time Hypothesis fails. Our reduction implies the hardness and lower bound for a general class of problems, where these problems come as a special case.

1 Introduction

For a finite set of graphs \mathcal{H} , \mathcal{H} -FREE EDGE DELETION problem asks whether we can delete at most k edges from an input graph G to obtain a graph G' such that for every $H \in \mathcal{H}$, G' does not have an induced copy of H . If $\mathcal{H} = \{H\}$, the problem is denoted by H -FREE EDGE DELETION. EDITING problems are defined similarly where we are allowed to add or delete at most k edges. \mathcal{H} -FREE EDGE DELETION comes under the broader category of graph modification problems which have found applications in DNA physical mapping [4], circuit design [12] and machine learning [3]. Cai has proved that \mathcal{H} -FREE EDGE DELETION is fixed parameter tractable [5]. Polynomial kernelization and incompressibility of these problems were subjected to rigorous studies in the recent past. Kratsch and Wahlström gave the first example on the incompressibility of H -FREE EDGE DELETION problems by proving that the problem is incompressible if H is a certain graph

^{*} A preliminary version of this paper has appeared in the proceedings of IPEC 2015.

^{**} supported by TCS Research Scholarship

on seven vertices, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ [16]. Later, it has been proved that there exist no polynomial kernel for H -FREE EDGE DELETION where H is any 3-connected graph other than a complete graph, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ [6]. In the same paper, under the same assumption, it is proved that, if H is a path or a cycle, then H -FREE EDGE DELETION is incompressible if and only if H has at least four edges. Except for a few cases, the kernelization complexity of H -FREE EDGE DELETION is known when H is a tree [7]. It has been proved that \mathcal{H} -FREE EDGE DELETION admits polynomial kernelization on bounded degree graphs if \mathcal{H} is a finite set of graphs [10]. Though kernelization complexities of many H -FREE EDGE DELETION problems are known, CLAW-FREE EDGE DELETION withstood the test of time and yielded neither an incompressibility result nor a polynomial kernel. Some progress has been made recently for this problem such as a polynomial kernel for CLAW-FREE EDGE DELETION on K_t -free input graphs [2] and a polynomial kernel for $\{\text{CLAW}, \text{DIAMOND}\}$ -FREE EDGE DELETION [9].

Motivation

As described above, the kernelization complexity of H -FREE EDGE DELETION is known when H is any 3-connected graph, path or cycle. Except for a few cases, the status is known when H is any tree. Every new insight into these problems may help us to obtain a dichotomy on the kernelization complexities of H -FREE EDGE DELETION problems.

The polynomial kernelization in this paper is inspired by two properties related to diamond graph. Firstly, a graph is diamond-free if and only if every edge is part of exactly one maximal clique. The second property is that, the neighborhood of every vertex in a diamond graph is connected. It can be easily verified that, when a graph H has this property and if H is diamond-free then H is a disjoint union of cliques (cluster). Though our kernelization technique may give polynomial kernels for \mathcal{H} -FREE EDGE DELETION, such that \mathcal{H} contains diamond and every other $H \in \mathcal{H}$ is a cluster, it complicates the analysis of the kernel size. Hence we restrict the study to DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION.

Our Results

In this paper, we study the polynomial kernelization and hardness results of DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION, where $t \geq 4$ is any fixed integer. It has been proved that DIAMOND-FREE EDGE DELETION admits a kernel of $O(k^4)$ vertices [13]. We improve this result by giving a kernel of $O(k^3)$ vertices. A proper subset of the rules applied for DIAMOND-FREE EDGE DELETION gives us an $O(k^2)$ vertex kernel for $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION. We use vertex modulator technique, which was used recently to give a polynomial kernel for TRIVIAALLY PERFECT EDITING [11] and to obtain a polynomial kernel for $\{\text{CLAW}, \text{DIAMOND}\}$ -FREE EDGE DELETION [9]. We introduce a rule named as *vertex-split* which *splits* a vertex v into a set of independent vertices where each vertex in the set corresponds to a component in

the neighborhood of v . We prove that, this rule is safe for many other H -FREE EDGE DELETION problems.

For any fixed $s \geq 1$, an s -diamond is defined as the graph $K_2 \times (s+1)K_1$. When $s = 1$, we get a diamond graph. As part of a dichotomy result on the hardness of H -FREE EDGE DELETION and H -FREE EDGE EDITING problems, it has been proved that s -DIAMOND-FREE EDGE DELETION and s -DIAMOND-FREE EDGE EDITING are NP-complete and cannot be solved in parameterized subexponential time, unless Exponential Time Hypothesis (ETH) fails [1]. We improve these results by proving that s -DIAMOND-FREE EDGE DELETION and s -DIAMOND-FREE EDGE EDITING are NP-complete even on K_4 -free graphs and can be solved neither in subexponential time nor in parameterized subexponential time, unless ETH fails. Our reduction implies that these results are applicable for $\{s$ -DIAMOND, $K_t\}$ -FREE EDGE DELETION and $\{s$ -DIAMOND, $K_t\}$ -FREE EDGE EDITING for any fixed $s \geq 1$ and $t \geq 4$. Fellows et. al. have proved [13] the hardness of a similar kind of problems termed as s -EDGE OVERLAP DELETION (s -EDGE OVERLAP EDITING), where the objective is to delete (edit) at most k edges from the input graph such that every edge in the resultant graph is part of at most s maximal cliques. We observe that when $s = 1$, s -DIAMOND-FREE EDGE DELETION (s -DIAMOND-FREE EDGE EDITING) coincides with s -EDGE OVERLAP DELETION (s -EDGE OVERLAP EDITING).

1.1 Preliminaries

Graphs: Every graph considered here is simple, finite and undirected. For a graph G , $V(G)$ and $E(G)$ denote the vertex set and the edge set of G respectively. $N_G(v)$ denotes the (open) neighborhood of a vertex $v \in V(G)$, which is the set of vertices adjacent to v in G . We remove the subscript when there is no ambiguity about the underlying graph G . A graph $G' = (V', E')$ is called an induced subgraph of a graph G if $V' \subseteq V(G)$, $E' \subseteq E(G)$ and an edge $\{x, y\} \in E(G)$ is in E' if and only if $\{x, y\} \subseteq V'$. For a vertex set $V' \subseteq V(G)$, $G[V']$ denotes the induced subgraph with a vertex set V' of G . A component G' of a graph G is a connected induced subgraph of G such that there is no edge between $V(G')$ and $V(G) \setminus V(G')$. For a set of vertices $V' \subseteq V(G)$, $G - V'$ denotes the graph obtained by removing the vertices in V' and all its incident edges from G . For an edge set $E' \subseteq E(G)$, $G - E'$ denotes the graph obtained by deleting all edges in E' from G . A matching (non-matching) is a set of edges (non-edges) such that every vertex in the graph is incident to at most one edge (non-edge) in the matching (non-matching). K_t denotes the complete graph on t vertices and $K_{1,s}$ denotes the graph where a vertex is adjacent to an independent set of s vertices, i.e., the graph $K_1 \times sK_1$. An s -diamond is defined as the graph $K_2 \times (s+1)K_1$ [1]. We note that 1-diamond is a diamond graph (see Figure 1). The edge between the two vertices with degree three in a diamond is the *middle edge* of the diamond. $K_{1,3}$ is also known as a claw graph. In this paper, \mathcal{H} always denotes a finite set of graphs. A graph G is called \mathcal{H} -free, if G does not contain any induced copy of any $H \in \mathcal{H}$.

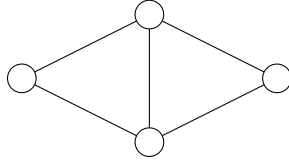


Fig. 1: 1-diamond is isomorphic to a diamond graph

Parameterized complexity: A parameterized problem is *fixed parameter tractable*, if there is an algorithm to solve it in time $f(k) \cdot n^{O(1)}$, where f is any computable function and n is the size of the input, and k is the parameter. A *polynomial kernelization* is an algorithm which takes as input (G, k) of a parameterized problem, runs in time $(|G| + k)^{O(1)}$ and returns an instance (G', k') of the same problem such that $|G'|, k' \leq p(k)$, where p is any polynomial function. A rule for kernelization is *safe* if (G, k) is a yes-instance if and only if (G', k') is a yes-instance where (G, k) and (G', k') are the input and output of the rule. A *linear reduction* is a polynomial time reduction from a problem A to another problem B such that $|G'| = O(|G|)$, where G and G' are the input and output of the reduction. A *linear parameterized reduction* from a parameterized problem A to another problem B is a polynomial time reduction such that $k' = O(k)$ where k and k' are the parameters of the instances of A and B respectively. A problem is solvable in *subexponential* time if it admits an algorithm which runs in time $2^{o(|G|)}$, where G is the input. Similarly, a parameterized problem is solvable in *parameterized subexponential* time if it admits an algorithm which runs in time $2^{o(k)} \cdot |G|^{O(1)}$, where G is the input and k is the parameter.

Exponential Time Hypothesis (ETH) (along with Sparsification Lemma [14]) implies that there is no algorithm which solves 3-SAT in time $2^{o(n+m)}$, where n is the number of variables and m is the number of clauses in the input instance. We can use a linear reduction from a problem (which does not admit subexponential time algorithm, assuming ETH) to another problem to show that the latter does not have a subexponential time algorithm, unless ETH fails. Similarly, under the same assumption, we can use a linear parameterized reduction from a parameterized problem (which does not admit parameterized subexponential time algorithm, assuming ETH) to another parameterized problem to show that the latter does not have a parameterized subexponential time algorithm. We refer the book [8] for further reading on parameterized algorithms and complexity.

2 Polynomial Kernels

In this section, we give a kernelization for DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION. There are two phases for the kernelization. In the first phase, apart from three standard rules, we introduce a new rule named as vertex-split, which has applications in the kernelization of other edge deletion problems. In the second phase, we apply vertex modulator technique.

2.1 Phase 1

We start with two standard rules. The first rule deletes an *irrelevant* edge and the second rule deletes a *must-delete* edge.

Definition 2.1 (Core Member) *Let G be an input graph of an \mathcal{H} -FREE EDGE DELETION problem. Then, a vertex or an edge of a graph G is a core member of G if it is contained in a subgraph (not necessarily induced) of G isomorphic to an $H \in \mathcal{H}$.*

Rule 1 (Irrelevant Edge) *Let G be an input to the rule, which is an input graph to an \mathcal{H} -FREE EDGE DELETION problem. If there is an edge $e \in E(G)$ which is not a core member of G , then delete e from G .*

Lemma 2.2 *Irrelevant edge rule is safe and can be applied in polynomial time for any \mathcal{H} -FREE EDGE DELETION.*

Proof. Let (G, k) be an instance of \mathcal{H} -FREE EDGE DELETION. Let G' be obtained by applying irrelevant edge rule on G . We claim that (G, k) is a yes-instance if and only if (G', k) is a yes-instance. Let S be a solution of size at most k of (G, k) . For a contradiction, assume that $G' - S$ has an induced $H \in \mathcal{H}$ with a vertex set D' . Since D' does not induce H in $G - S$, the edge e deleted by irrelevant edge rule has both the end points in D' . Then D' induces a supergraph of H in G , which is a contradiction. Conversely, let S' be a solution of size at most k of (G', k) . Assume that $G - S'$ has an induced $H \in \mathcal{H}$ with vertex set D . The edge deleted by irrelevant edge rule has both the end points in D . This implies that D induces a supergraph of H in G , which is a contradiction. Since, in polynomial time, we can verify whether an edge is part of an $H \in \mathcal{H}$ in G , the rule can be applied in polynomial time.

Corollary 2.3 *Irrelevant edge rule is safe and can be applied in polynomial time for DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION.*

The next rule deletes an edge e , if e is the middle edge of $k + 1$ otherwise edge-disjoint diamonds. This rule is found in [13].

Rule 2 (Sunflower) *Let (G, k) be an input to the rule. If there is an edge $e = \{x, y\} \in E(G)$ such that $G[N(x) \cap N(y)]$ has a non-matching of size at least $k + 1$, then delete e from G and decrease k by 1.*

Lemma 2.4 *Sunflower rule is safe and can be applied in polynomial time.*

Proof. Let (G, k) be an instance of DIAMOND-FREE EDGE DELETION ($\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION). Let $e = \{x, y\} \in E(G)$ and V' be $N(x) \cap N(y)$. Assume that $G[V']$ has a non-matching M' of size at least $k + 1$. Let sunflower rule be applied on (G, k) to obtain $(G - e, k - 1)$. It is enough to prove that every solution S of size at most k of (G, k) contains the edge e . Every non-edge

$\{a, b\}$ in M' corresponds to an induced diamond $\{x, y, a, b\}$ in G . The diamonds corresponds to any two different non-edges in M' share only one edge $\{x, y\}$. Since at least one edge from every induced diamond is in S , e must be in S . The rule can be applied in polynomial time as maximum non-matching can be found in polynomial-time.

Now we introduce a property and a rule based on it.

Definition 2.5 (Connected Neighborhood) *For a graph G and a vertex $v \in V(G)$, v has connected neighborhood if $G[N(v)]$ is connected. G has connected neighborhood if every vertex in G has connected neighborhood.*

Rule 3 (Vertex-Split) *Let $v \in V(G)$ and v does not have connected neighborhood in G . Let there be $t > 1$ components in $G[N(v)]$ with vertex sets V_1, V_2, \dots, V_t . Introduce t new vertices v_1, v_2, \dots, v_t and make v_i adjacent to all vertices in V_i for $1 \leq i \leq t$. Delete v .*

An example of the application of vertex-split rule is depicted in Figure 2. We denote the set of vertices created by splitting v by V_v . Let G' be the graph obtained by splitting a vertex v in G . For convenience, we identify an edge (v, u) in G with an edge (v_j, u) in G' where u is in the j^{th} component of $G[N(v)]$, so that for every set of edges S in G , there is a corresponding set of edges in G' and vice versa. We identify a set of vertices $V' \subseteq V(G) \setminus \{v\}$ with the corresponding vertices in G' . Similarly, we identify $V' \subseteq V(G') \setminus V_v$ with the corresponding vertex set in G . Before proving the safety of the rule, we prove two simple observations.

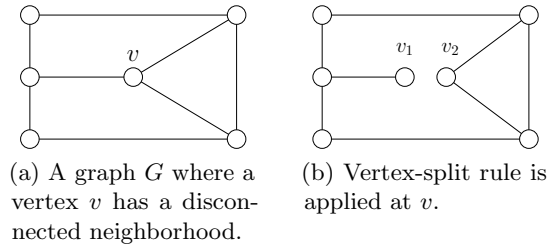


Fig. 2: An application of vertex-split rule

Observation 2.6 *Let vertex-split rule be applied on G to obtain G' . Let $v \in V(G)$ be the vertex being split.*

- (i) *Then, for every pair of vertices $\{v_i, v_j\} \subseteq V_v$, the distance between v_i and v_j is at least four.*

- (ii) Let $u \in V(G) \setminus \{v\}$ and u has connected neighborhood in G . Then u has connected neighborhood in G' . Furthermore, every new vertex v_i introduced in G' has connected neighborhood.

Proof. (i). Let $\{v_i, v_j\} \subseteq V_v$. Clearly, v_i and v_j are non-adjacent. Consider any two vertices $u_i \in N(v_i)$ and $u_j \in N(v_j)$. If $u_i = u_j$ or u_i and u_j are adjacent in G' , there would be only one vertex generated for the component containing u_i and u_j in $G[N(v)]$ by splitting v , which is a contradiction. It follows that the distance between v_i and v_j is at least four.

(ii). If $v \notin N_G(u)$, then the neighborhood of u is not disturbed by the rule and hence u has connected neighborhood in G' . Let $v \in N(u)$. Let v_i be the vertex generated by splitting v for the component in $G[N(v)]$ containing u . Since, there is only one new vertex introduced for a component of $G[N(v)]$, no other new vertex is adjacent to u in G' . It is straight-forward to verify that v_i in $G'[N_{G'}(u)]$ plays the role of v in $N_G(u)$ and hence $G[N_G(u)]$ and $G'[N_{G'}(u)]$ are isomorphic. To prove the last statement, we observe that, since a new vertex is made adjacent to a component in the neighborhood of v , every new vertex v_j in G' has connected neighborhood.

Lemma 2.7 *Vertex-split rule is safe and can be applied in polynomial time for any \mathcal{H} -FREE EDGE DELETION problem where every $H \in \mathcal{H}$ has diameter at most two and has connected neighborhood.*

Proof. Let G' be obtained by applying vertex-split rule on a vertex v of G . We claim that (G, k) is a yes-instance if and only if (G', k) is a yes-instance.

Let (G, k) be a yes-instance. Let S be a solution of size at most k of (G, k) . For a contradiction, assume that $G' - S$ has an induced $H \in \mathcal{H}$ with a vertex set D' . Since $G - S$ is H -free, D' must contain at least one newly created vertex v_i . Since the diameter of H is at most two, by Observation 2.6(i), D' can contain at most one newly created vertex. Hence, let $D' \cap V_v = \{v_i\}$. Since H is connected, $D' \cap N(v) \neq \emptyset$. Now, there are two cases and in each case we get a contradiction.

- (i) $D' \cap N(v) \subseteq V_i$: In this case, v_i plays the role of v and hence $(G - S)[(D' \setminus v_i) \cup \{v\}]$ and $(G' - S)[D']$ are isomorphic.
- (ii) D' contains vertices from multiple components of $N(v)$, i.e., $(D' \cap N(v)) \setminus V_i \neq \emptyset$: Let $u_j \in D' \cap (N(v) \setminus V_i)$. Now, it is straight-forward to verify that the distance between v_i and u_j is at least three in $(G' - S)[D']$, which is a contradiction to the fact that H has diameter at most two.

For the converse, let S' be a solution of size at most k of (G', k) . For a contradiction, assume that $G - S'$ has an induced $H \in \mathcal{H}$ with a vertex set D . Clearly, $v \in D$. Since H is connected, there exists a $u_i \in D \cap N(v)$. Now there are two cases and in each case we get a contradiction.

- (i) $D \cap N(v) \subseteq V_i$: In this case, $(D \setminus \{v\}) \cup \{v_i\}$ induces H in $G' - S'$, which is a contradiction.

- (ii) D contains vertices from multiple components of $N(v)$: Let $u_j \in (D \cap N(v)) \setminus V_i$. Here, it can be verified that either v does not have connected neighborhood in $(G - S')[D]$ or the distance between either v and u_i or between v and u_j is at least three, which is a contradiction.

It is straight-forward to verify that splitting a vertex can be done in linear time.

Corollary 2.8 *Vertex-split rule is safe and can be applied in polynomial time for the problems DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION.*

The next rule is a trivial one and the safety of it can be easily verified.

Rule 4 (Irrelevant component) *Let G be an input to the rule, which is an input graph of an \mathcal{H} -FREE EDGE DELETION problem. If a component of G is \mathcal{H} -free, then delete the component from G .*

Lemma 2.9 *Irrelevant component rule is safe and can be applied in polynomial time for every \mathcal{H} -FREE EDGE DELETION problem.*

Corollary 2.10 *Irrelevant component rule is safe and can be applied in polynomial time for DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION.*

Now, we are ready with the Phase 1 of the kernelization for DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION.

Phase 1

Let (G, k) be an input to Phase 1.

- Exhaustively apply rules irrelevant edge, sunflower, vertex split and irrelevant component on (G, k) to obtain (G', k) .

Lemma 2.11 *Let (G, k) be an instance of DIAMOND-FREE EDGE DELETION ($\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION). Let (G', k') be obtained by applying Phase 1 on (G, k) . Then:*

- (i) *Every vertex and edge in G is a core member.*
- (ii) *G' has connected neighborhood.*
- (iii) *$|E(G')| \leq |E(G)|$ and $|V(G')| \leq 2|E(G)|$.*

Proof. (i) and (ii) follow from the fact that irrelevant edge and vertex-split rules are not applicable on (G', k') . (iii) follows from the fact that none of the rules increases the number of edges in the graph.

Lemma 2.12 *Applying Phase 1 is safe and Phase 1 runs in polynomial time.*

Proof. The safety follows from the safety of the rules being applied. Single application of each rule can be done in polynomial time (Corollary 2.3, Lemma 2.4, Corollary 2.8 and Corollary 2.10). None of the rules increases the number of edges. Hence, number of applications of irrelevant edge rule and sunflower rule is linear. An application of irrelevant component rule does not necessitate an application of vertex-split rule. By Observation 2.6(ii), an application of vertex-split rule decreases the number of vertices with disconnected neighborhood. Hence, between two applications of either irrelevant edge rule or sunflower rule, only a linear number of applications of vertex-split rule is possible. Hence, vertex-split rule can be applied only polynomial number of times. Since, only vertex-split rule increases the number of vertices and there are only polynomial many applications of it, irrelevant-component rule can be applied only polynomial number of times.

2.2 Phase 2

In this phase, we apply vertex modulator technique to complete the kernelization of DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION. We define a vertex modulator for DIAMOND-FREE EDGE DELETION ($\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION) similar to that defined for TRIVIAALLY PERFECT EDITING [11].

Definition 2.13 (D-modulator) *Let (G, k) be an instance of DIAMOND-FREE EDGE DELETION ($\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION). Let $V' \subseteq V(G)$ be such that $G - V'$ is diamond-free ($\{\text{diamond}, K_t\}$ -free). Then, V' is called a D-modulator.*

Now we state a folklore characterization of diamond-free graphs.

Proposition 2.14 *A graph G is diamond-free if and only if every edge in G is a part of exactly one maximal clique.*

For a diamond-free graph G , since every edge is in exactly one maximal clique, there is a unique way of partitioning the edges into maximal cliques. For convenience, we call the set of subsets of vertices, where each subset is the vertex set of a maximal clique, as a *maximal clique partitioning*. We note that, one vertex may be a part of many sets in the partitioning.

Lemma 2.15 *Let (G, k) be an instance of DIAMOND-FREE EDGE DELETION ($\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION). Then, in polynomial time, the edge set X of size at most $5k(t \cdot (t-1) \cdot k/2)$ of a maximal set of edge-disjoint diamonds (diamonds and K_t s), a D-modulator V_X of size at most $4k(tk)$ and a maximal clique partitioning \mathcal{C} of $G - V_X$ can be obtained or it can be declared that (G, k) is a no-instance.*

Proof. We prove the lemma for DIAMOND-FREE EDGE DELETION. Similar arguments apply for $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION. Let $X = \emptyset$. Include edges of any induced diamond of G in X . Then, iteratively include edges of any

induced diamond of $G - X$ in X until $k + 1$ iterations are completed or no more induced diamond is found in $G - X$. If $k + 1$ iterations are completed, then we can declare that the instance is a no-instance as every solution must have at least one edge from every induced diamonds. If the number of iterations is less than $k + 1$ such that there is no induced diamond in $G - X$, then $|X| \leq 5k$, as every diamond has five edges. Let V_X be the set of vertices incident to the edges in X . Then $|V_X| \leq 4k$, as every diamond has four vertices. Since $G - V_X$ has no induced diamond, V_X is a D-modulator. Since, there are only at most $k + 1$ iterations and each iteration takes polynomial time, this can be done in polynomial time. Since $G - V_X$ is diamond-free, by Proposition 2.14, every edge in it is part of exactly one maximal clique. Now, the maximal clique partitioning \mathcal{C} of $G - V_X$ where each $C \in \mathcal{C}$ is a set of vertices of a maximal clique, can be found by greedily obtaining the maximal cliques, which can be done in polynomial time.

Let (G, k) be an output of Phase 1. Here onward, we assume that X is an edge set of the maximal set of edge-disjoint diamonds (diamonds and K_t s), V_X is a D-modulator, which is the set of vertices incident to X and \mathcal{C} is the maximal clique partitioning of $G - V_X$. Observation 2.16 directly follows from the maximality of X . Observation 2.17 is found in Lemma 3.1 of [9]. It was proved there, if G is $\{\text{claw, diamond}\}$ -free, but is also applicable if G is diamond-free.

Observation 2.16 *Every induced diamond (diamond and K_t) in G has an edge in X .*

Observation 2.17 *Let $C, C' \in \mathcal{C}$ and be distinct. Then:*

- (i) $|C \cap C'| \leq 1$.
- (ii) *If $v \in C \cap C'$, then there is no edge between $C \setminus \{v\}$ and $C' \setminus \{v\}$.*

Proof. (i). Assume that $x, y \in C \cap C'$. Then the edge $\{x, y\}$ is part of two maximal cliques, which is a contradiction by Proposition 2.14.

(ii). Let $x \in C \setminus \{v\}$ and $y \in C' \setminus \{v\}$. Let x and y be adjacent. Clearly, $\{x, y\}$ is not part of the clique induced by C . Now, $\{x, v\}$ is part of not only the clique induced by C but also a maximal clique containing x, y and v , which is a contradiction.

Definition 2.18 (Local Vertex) *Let G be a graph and $C \subseteq V(G)$ induces a clique in G . A vertex v in C is called local to C in G , if $N(v) \subseteq C$.*

Lemma 2.19 *Let (G, k) be an instance of DIAMOND-FREE EDGE DELETION ($\{\text{DIAMOND, } K_t\}$ -FREE EDGE DELETION). Let C be a clique with at least $2k + 2$ vertices in G .*

- (i) *Every solution S of size at most k of (G, k) does not contain any edge e where both the end points of e are in C .*
- (ii) *Let $C' \subseteq C$ be such that every vertex $v \in C'$ is local to C in G . Every induced diamond with vertex set D in G can contain at most one vertex in C' .*

Proof. (i). Let $e = \{x, y\}$ be an edge in G such that $x, y \in C$. Let S be a solution of size at most k of (G, k) such that $e \in S$. Consider any two vertices $a, b \in C \setminus \{x, y\}$ (assuming k is at least 1). Clearly, $\{a, b, x, y\}$ induces a diamond in $G - e$. Consider a maximum matching M of $G[C \setminus \{x, y\}]$. Since $C \setminus \{x, y\}$ induces a clique of size at least $2k$ in G , $|M| \geq k$. For any two edges $\{a, b\}, \{a', b'\} \in M$, the diamonds induced by $\{a, b, x, y\}$ and $\{a', b', x, y\}$ are edge-disjoint. S must contain one edge from the diamonds corresponds each edge in M . Since $e \in S$, $|S| \geq k + 1$, which is a contradiction.

(ii). For a contradiction, assume that D induces a diamond in G and D contains two vertices $\{x, y\}$ of C' . Let a and b be the other two vertices in D . Since x and y are local to C in G , $a, b \in C$. Hence, $\{a, b, x, y\}$ is a clique in G , which is a contradiction.

We note that the following lemma is applicable only for DIAMOND-FREE EDGE DELETION.

Lemma 2.20 *Let $C' \subseteq C$ be such that every vertex $v \in C'$ is local to C in G . Then, for the problem DIAMOND-FREE EDGE DELETION, it is safe to delete $\min\{|C'| - 1, |C| - (2k + 2)\}$ vertices of C' in G .*

Proof. Let G' be obtained by deleting a set C'' of t vertices of C' from G such that $t = \min\{|C'| - 1, |C| - (2k + 2)\}$. We need to prove that (G, k) is a yes-instance if and only if (G', k) is a yes-instance. Let S be a solution of size at most k of (G, k) . Since $G' - S$ is an induced subgraph of $G - S$, and $G - S$ is diamond-free, we obtain that $G' - S$ is diamond-free. Conversely, let S' be a solution of size at most k of (G', k) . We claim that S' is a solution of (G, k) . Assume not. Let $G - S'$ has an induced diamond with a vertex set D . Since $|C| - |C''| \geq 2k + 2$, by 2.19(i), S' does not contain any edge in the clique induced by $C \setminus C''$ in G' . Now there are three cases:

- (a). $C'' \cap D = \emptyset$: In this case D induces a diamond in $G' - S'$, which is a contradiction.
- (b). $C'' \cap D = \{v\}$: We observe that we retained at least one vertex u of C' in G' . By 2.19(ii), D does not contain any other vertex from C' . Then, $D \cup \{u\} \setminus \{v\}$ induces a diamond in $G' - S'$.
- (c). $|C'' \cap D| \geq 2$: This case is not possible by 2.19(ii).

We define $\mathcal{C}_i \subseteq \mathcal{C}$ as the set of sets of vertices of the maximal cliques with exactly i vertices. Similarly, $\mathcal{C}_{\geq i} \subseteq \mathcal{C}$ denote the set of sets of vertices of the maximal cliques with at least i vertices.

The first in the following observation has been proved in Lemma 3.2 in [9] in the context where $G - V_X$ is $\{\text{diamond}, \text{claw}\}$ -free. Here we prove it in the context where $G - V_X$ is diamond-free.

Observation 2.21 *Let $C \in \mathcal{C}$. Then:*

- (i) *If there is a vertex $v \in V_X$ such that v is adjacent to at least two vertices in C , then v is adjacent to all vertices in C .*

(ii) A vertex in $V(G) \setminus (V_X \cup C)$ is adjacent to at most one vertex in C .

Proof. (i). Let v is adjacent to two vertices x, y in C but not adjacent to $z \in C$. Then $\{x, y, v, z\}$ induces a diamond such that none of the edges of the diamond is in X .

(ii). Assume that a vertex $u \in V(G) \setminus (V_X \cup C)$ is adjacent to all vertices in C . This contradicts with the fact that C induces a maximal clique in $G - V_X$. Let u be adjacent to at least two vertices $\{a, b\}$ in C and non-adjacent to at least one vertex $v \in C$. Then $\{a, b, u, v\}$ induces a diamond where none of the edges of the diamond is in X .

Consider $C \in \mathcal{C}$. We define three sets of vertices in G based on C .

$$A_C = \{v \in V_X : v \text{ is adjacent to all vertices in } C\}$$

$$B_C = \{v \in V(G) \setminus (V_X \cup C) : v \text{ is adjacent to exactly one vertex in } C\}$$

$$D_C = \{v \in V_X : v \text{ is adjacent to exactly one vertex in } C\}$$

For a vertex $v \in C$, let B_v denote the set of all vertices in B_C adjacent to v . Similarly let D_v denote the set of all vertices in D_C adjacent to v .

Observation 2.22 *Let $C \in \mathcal{C}$. Then,*

- (i) *The set of vertices in $V(G) \setminus C$ adjacent to at least one vertex in C is $A_C \cup B_C \cup D_C$.*
- (ii) *If $|C| > 1$, then A_C induces a clique in G .*
- (iii) *For two vertices $u, v \in C$, $B_u \cap B_v = \emptyset$ and $D_u \cap D_v = \emptyset$.*

Proof. (i) directly follows from Observation 2.21.

(ii). Assume not. Let a and b be two non-adjacent vertices in A_C . By Observation 2.21(i), both a and b are adjacent to all vertices in C . Consider any two vertices $x, y \in C$. $\{x, y, a, b\}$ induces a diamond with no edge in X , which is a contradiction.

(iii) directly follows from the definition of B_C and D_C .

Lemma 2.23 *Let $v \in C \in \mathcal{C}$. If B_v is non-empty then D_v is non-empty.*

Proof. The statement is trivially true if $|C| = 1$. Hence assume that $|C| \geq 2$. Since v has connected neighborhood, $G[N(v)]$ is connected. We observe that $N(v) = A_C \cup B_v \cup D_v \cup (C \setminus \{v\})$. Assume B_v is non-empty. By Observation 2.21(ii), there is no edge between the sets B_v and $C \setminus \{v\}$. Consider a vertex $v_b \in B_v$ adjacent to $A_C \cup D_v$. Assume v_b is not adjacent to D_v . Then v_b must be adjacent to a vertex $v_a \in A_C$. Let v' be any other vertex in C . Then $\{v_a, v, v', v_b\}$ induces a diamond which has no edge intersection with X . Therefore v_b must be adjacent to a vertex in D_v .

Observation 2.24 *Let $C \in \mathcal{C}$. Then there are two adjacent vertices x and y such that $x \in A_C$ and $y \in A_C \cup D_C$.*

Proof. **Case 1:** $C = \{v\} \in \mathcal{C}_1$. Since $\{v\} \in \mathcal{C}_1$, v is not adjacent to any vertex in $V(G) \setminus V_X$. Since v is a core member, v is part of an induced diamond or K_4 in G . Hence there exist two adjacent vertices $x, y \in A_C$.

Case 2: $|C| \geq 2$. Assume that $|A_C| = 0$. If $B_C \cup D_C = \emptyset$, then by Observation 2.22(i), the clique C is a component in G . Then, irrelevant component rule is applicable. Hence $B_C \cup D_C$ is non-empty. Consider a vertex $v \in C$ such that $B_v \cup D_v$ is non-empty. By Observation 2.22(iii), $B_v \cup D_v$ is not adjacent to any vertex in $C \setminus \{v\}$. Hence, $G[N(v)]$ has at least two components, one from $B_v \cup D_v$ and the other from C , which contradicts with the fact that v has connected neighborhood. Hence, $|A_C| > 0$. Assume $|A_C = \{x\}| = 1$. For a contradiction, assume that $D_C = \emptyset$. Then Lemma 2.23 implies that B_C is empty. Then x does not have connected neighborhood or $C \cup \{x\}$ induces an irrelevant component, which are contradictions. Hence, D_C is non-empty. If $|A_C| \geq 2$, then we are done by Observation 2.22(ii).

Lemma 2.25 *In the context of DIAMOND-FREE EDGE DELETION, let $C \in \mathcal{C}_{\geq 3}$. Then, the number of vertices in C which are adjacent to at least one vertex in $B_C \cup D_C$ is at most $4k - 1$.*

Proof. By Observation 2.24, $|A_C| \geq 1$. Since $|V_X| \leq 4k$, $|D_C| \leq 4k - 1$. Let C' be the set of vertices in C which are adjacent to $B_C \cup D_C$. For every vertex $v \in C'$, by Lemma 2.23, if B_v is non-empty, then D_v is non-empty. Since $v \in C'$, if B_v is empty, then also D_v is non-empty. For any two vertices $v, u \in C'$, by Observation 2.22(iii), $D_u \cap D_v = \emptyset$. Therefore $|C'| \leq |D_C| \leq 4k - 1$.

Now, we state the last rule of the kernelization. We apply this rule only for DIAMOND-FREE EDGE DELETION.

Rule 5 (Clique Reduction) *Let $C \in \mathcal{C}_{\geq 3}$ be such that $|C| > 4k$. Let C' be $C \cup A_C$. Let C'' be the set of vertices in C which are local to C' . Then, delete any $|C''| - 1$ vertices from C'' .*

Clique reduction rule helps us to reduce the size of large cliques in the clique partitioning in the context of DIAMOND-FREE EDGE DELETION. This rule is not required for $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION as the size of the cliques in the clique partitioning is already bounded.

Observation 2.26 *After the application of clique reduction rule, the number of vertices retained in C is at most $4k$.*

Proof. By Lemma 2.25, the number of vertices in C which are not local to C' is at most $4k - 1$. Hence, the rest of the vertices in C are local to C' in G . If $|C| > 4k$, clique reduction rule retains only one local vertex and delete all other vertices in C local to C' .

Lemma 2.27 *Clique reduction rule is safe for DIAMOND-FREE EDGE DELETION and can be applied in polynomial time.*

Proof. The safety of the rule follows from Lemma 2.20. It is straight-forward to verify that the rule can be applied in polynomial-time.

Now we give the kernelization algorithms.

Kernelization of DIAMOND-FREE EDGE DELETION

Let (G, k) be the input.

Step 1: Apply Phase 1 on (G, k) to obtain (G_1, k') .

Step 2: Greedily pack edge disjoint diamonds of (G_1, k') . If the count of edge disjoint diamonds in the pack exceeds k , then declare that the instance is a no-instance; Otherwise find X, V_X and \mathcal{C} of G_1 as given in Lemma 2.15 from the maximal greedy packing.

Step 3: Exhaustively apply clique reduction rule on (G_1, k') to obtain (G', k_1) .

Kernelization of $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION

Let (G, k) be the input. Apply Phase 1 on (G, k) and return the output (G', k') . If the size of a greedy packing of edge disjoint diamonds and K_t s of G' exceeds k , then declare that the instance is a no-instance; Otherwise return (G', k') .

Lemma 2.28 *The kernelization algorithms for DIAMOND-FREE EDGE DELETION and $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION are safe and can be applied in polynomial time.*

Proof. The safety of the kernelizations follow directly from the safety of Phase 1 and clique reduction rule (Lemma 2.12, 2.27). Every application of clique reduction rule decreases the number of edges. Hence by Lemma 2.12, 2.15 and 2.27 both the kernelization runs in polynomial time.

2.3 Bounding the Kernel Size

In this subsection, we bound the number of vertices in the kernels obtained by the kernelizations. Let (G, k) be an instance of DIAMOND-FREE EDGE DELETION ($\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION) and (G', k') is obtained by the kernelization. Consider an X, V_X and \mathcal{C} of (G', k') as obtained by Lemma 2.15 in the case of $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION and that obtained after the last application of clique reduction rule in the case of DIAMOND-FREE EDGE DELETION.

Lemma 2.29 $\sum_{C \in \mathcal{C}_1} |C| = O(k^2)$.

Proof. Let $\{v\} \in \mathcal{C}_1$. By Observation 2.24, v must be adjacent to two vertices $x, y \in V_X$ such that x and y are adjacent. Consider an edge $\{x, y\} \in X$. In the common neighborhood of $\{x, y\}$ there can be at most $2k + 1$ vertices v with the property that $\{v\} \in \mathcal{C}_1$, otherwise sunflower rule applies. Now, consider an edge $\{x, y\} \in E(G'[V_X] - X)$. In the common neighborhood of $\{x, y\}$ there can be at

most one vertex v with the property that $\{v\} \in \mathcal{C}_1$, otherwise there is an induced diamond edge-disjoint with X . Since there are $O(k)$ edges in X and $O(k^2)$ edges in $E(G'[V_X] - X)$, we obtain the result.

Lemma 2.30 (i) Consider any two vertices $x, y \in V_X$. Let $\mathcal{C}' \subseteq \mathcal{C}_{\geq 2}$ such that for any $C \in \mathcal{C}'$, $x, y \in A_C$. If $\{x, y\} \in X$ then $|\mathcal{C}'| \leq 2k + 1$. If $\{x, y\} \notin X$, then $|\mathcal{C}'| \leq 1$.
(ii) Consider any ordered pair of vertices (x, y) in V_X such that x and y are adjacent in G' . Let $\mathcal{C}' \subseteq \mathcal{C}_{\geq 2}$ such that for any $C \in \mathcal{C}'$, $x \in A_C$ and $y \in D_C$. If $\{x, y\} \in X$ then $|\mathcal{C}'| \leq 2k + 1$. If $\{x, y\} \notin X$, then $|\mathcal{C}'| = 0$.

Proof. (i). Let $C_a, C_b \in \mathcal{C}'$. By Observation 2.17(i), $|C_a \cap C_b| \leq 1$. If $v \in C_a \cap C_b$, then by Observation 2.17(ii), there is no edge between $C_a \setminus \{v\}$ and $C_b \setminus \{v\}$. Hence, $\{x, v, a, b\}$ induces a diamond where $a \in C_a \setminus \{v\}$ and $b \in C_b \setminus \{v\}$, which is edge disjoint with X , a contradiction. Hence $C_a \cap C_b = \emptyset$. Now, consider any two vertices $a \in C_a$ and $b \in C_b$. Clearly, $\{x, y, a, b\}$ induces a diamond. Hence, $\{x, y\}$ must be an edge in X , otherwise the diamond is edge disjoint with X , a contradiction. Therefore, if $\{x, y\} \notin X$, $|\mathcal{C}'| \leq 1$. Now we consider the case in which $\{x, y\} \in X$. If $|\mathcal{C}'| \geq 2k + 2$, we get at least $k + 1$ diamonds where every two diamonds have the only edge intersection $\{x, y\}$. Then sunflower rule applies, which is a contradiction.

(ii). Let \mathcal{C}' be the set of all $C \in \mathcal{C}_{\geq 2}$ such that $x \in A_C$ and $y \in D_C$. Consider any two of them - C_a and C_b . By Observation 2.17(i), $|C_a \cap C_b| \leq 1$. If $v \in C_a \cap C_b$, then by Observation 2.17(ii), there is no edge between $C_a \setminus \{v\}$ and $C_b \setminus \{v\}$. Let $a \in C_a \setminus \{v\}$ and $b \in C_b \setminus \{v\}$. Then $\{x, v, a, b\}$ induces a diamond which is edge disjoint with X , a contradiction. Hence $C_a \cap C_b = \emptyset$. Let $a, a' \in C_a$ such that a is adjacent to y . Then, if $\{x, y\} \notin X$, $\{x, a, a', y\}$ induces a diamond, which is edge disjoint with X . Therefore, if $\{x, y\} \notin X$, then $|\mathcal{C}'| = 0$. Now we consider the case in which $\{x, y\} \in X$. If $|\mathcal{C}'| \geq 2k + 2$, we get at least $k + 1$ diamonds where every two diamonds have the only edge intersection $\{x, y\}$. Then sunflower rule applies, which is a contradiction.

Lemma 2.31 (i) For DIAMOND-FREE EDGE DELETION, $\sum_{C \in \mathcal{C}_{\geq 2}} |C| = O(k^3)$.
(ii) For $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION, $\sum_{C \in \mathcal{C}_{\geq 2}} |C| = O(k^2)$.

Proof. (i). Consider any two adjacent vertices $x, y \in V_X$. Let $\mathcal{C}'_{xy} \subseteq \mathcal{C}_{\geq 2}$ be such that $x, y \in A_C$. Then by Lemma 2.30(i), if $\{x, y\} \in X$, then $|\mathcal{C}'_{xy}| \leq 2k + 1$ and if $\{x, y\} \notin X$, then $|\mathcal{C}'_{xy}| \leq 1$. Since there are at most $5k$ edges in X and $O(k^2)$ edges in $G[V_X] \setminus X$, $\bigcup_{\{x, y\} \in E(G[V_X])} \mathcal{C}'_{xy}$ has at most $O(k) \cdot (2k + 1) + O(k^2) = O(k^2)$ maximal cliques. Since every maximal clique has at most $4k$ vertices (by Observation 2.26), the total number of vertices in those cliques is $O(k^3)$.

Now, let $\mathcal{C}'_{xy} \subseteq \mathcal{C}_{\geq 2}$ be such that $x \in A_C$ and $y \in D_C$. Then by Lemma 2.30(ii), if $\{x, y\} \in X$, then $|\mathcal{C}'_{xy}| \leq 2k + 1$ and if $\{x, y\} \notin X$, then $|\mathcal{C}'_{xy}| = 0$. Since there are at most $2 \cdot 5k = 10k$ ordered adjacent pairs of vertices in X , $\bigcup_{\{x, y\} \in E(G[V_X])} \mathcal{C}'_{xy}$ has at most $O(k) \cdot (2k + 1)$ maximal cliques. Since every maximal clique has at

most $4k$ vertices (by Observation 2.26), the total number of vertices in those cliques is $O(k^3)$.

Since, by Observation 2.24, for every $C \in \mathcal{C}$, there exist two vertices $x \in A_C$ and $y \in A_C \cup D_C$, we have counted every $C \in \mathcal{C}_2 \cup \mathcal{C}_{\geq 3}$. Hence $\sum_{C \in \mathcal{C}_2 \cup \mathcal{C}_{\geq 3}} |C| = O(k^3)$.

(ii). Since $G - V_X$ is K_t -free, every maximal clique in \mathcal{C} has at most $t - 1$ vertices. Hence from the above arguments, we obtain the result.

Theorem 2.32 *Given an instance (G, k) of DIAMOND-FREE EDGE DELETION, the kernelization gives an instance (G', k') such that $|V(G')| = O(k^3)$ and $k' \leq k$ or declares that the instance is a no-instance. Similarly, given an instance (G, k) of $\{\text{DIAMOND}, K_t\}$ -FREE EDGE DELETION, the kernelization gives an instance (G', k') such that $|V(G')| = O(k^2)$ and $k' \leq k$ or declare that the instance is a no-instance.*

Proof. None of the rules increases the parameter k . Then, the theorem follows from Lemma 2.29 and Lemma 2.31 and the fact that $|V_X| = O(k)$.

3 Hardness Results

In this section we prove that, for any fixed $s \geq 1$, s -DIAMOND-FREE EDGE DELETION is NP-complete even for K_4 -free graphs. As a corollary, we obtain that, for any fixed $s \geq 1$ and $t \geq 4$, $\{s\text{-DIAMOND}, K_t\}$ -FREE EDGE DELETION is NP-complete. We also obtain that these NP-complete problems can be solved neither in subexponential time nor in parameterized subexponential time, unless ETH fails. Further, we obtain similar results for s -DIAMOND-FREE EDGE EDITING and $\{s\text{-DIAMOND}, K_t\}$ -FREE EDGE EDITING.

It is known that VERTEX COVER is NP-complete on sub-cubic graphs [17]. It is also known that VERTEX COVER is NP-complete on triangle-free graphs [19, 20] by the simple observation that a graph G has a vertex cover of size at most k if and only if the graph obtained from G by sub-dividing every edge twice has a vertex cover of size at most $k + |E(G)|$. Combining these two reductions implies that VERTEX COVER is NP-complete on triangle-free sub-cubic graphs. Recently, Komusiewicz generalized this technique to obtain a general result, where the result for triangle-free sub-cubic graphs comes as a special case [15]. It also gives that VERTEX COVER on triangle-free sub-cubic graphs cannot be solved in subexponential time, unless ETH fails.

Proposition 3.1 [15] *VERTEX COVER on triangle-free sub-cubic graphs is NP-complete. Further, the problem cannot be solved in time $2^{o(|G|)}$, unless ETH fails.*

Since, the reduction in [15] is not a linear parameterized reduction, we need to compose it with the reduction from 3-SAT to VERTEX COVER to obtain a linear parameterized reduction.

Lemma 3.2 VERTEX COVER on triangle-free sub-cubic graphs cannot be solved in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.

Proof. The reduction from 3-SAT to VERTEX COVER on sub-cubic graphs (see [17]) gives an instance (G, k) of VERTEX COVER where G is sub-cubic and has $9m$ vertices and $12m$ edges and $k = 5m$, where m is the number of clauses of the input 3-SAT instance. Now, replace every edge of G by a path of three edges (i.e., subdivide every edge twice) to obtain G' . Now, G' has $33m$ vertices and $36m$ edges. It is straight forward to verify that the input 3-SAT instance is satisfiable if and only if G' has a vertex cover of size at most $17m$. Now, the statement follows from the observation that the reduction is linear parameterized.

Komusiewicz has also proved that [15] for any non-trivial hereditary property Π , Π VERTEX DELETION cannot be solved in time $2^{o(|G|)}$, unless ETH fails. The reduction is from a variant of VERTEX COVER. We require only the case when Π is ' $K_{1,s}$ -free'. For the sake of completeness, we reiterate his proof tailored for our special case.

Proposition 3.3 [15] For any fixed $s \geq 2$, $K_{1,s}$ -FREE VERTEX DELETION on triangle-free graphs with maximum degree at most $s + 2$ is NP-complete and can be solved neither in time $2^{o(|G|)}$ nor in time $2^{o(k)} \cdot |G|^{O(1)}$.

Proof. Let (G, k) be an instance of VERTEX COVER on triangle-free sub-cubic graphs. For every vertex v_i in G , introduce an independent set I_i of $s - 1$ vertices and make all of them adjacent to v_i . Let the resultant graph be G' . Clearly, G' is triangle-free and has degree at most $s + 2$.

Let S be a vertex cover of size at most k of G . Since $G - S$ is an independent set, $G' - S$ is a graph with degree at most $s - 1$ and hence $K_{1,s}$ -free. Conversely, let $S' \subseteq V(G')$ be such that $|S'| \leq k$ and $G' - S'$ is $K_{1,s}$ -free. For every vertex $u_i \in I_i$ such that $u_i \in S'$, replace u_i by v_i to obtain S'' . Clearly, $G' - S''$ is $K_{1,s}$ -free and $|S''| \leq k$. It is straight forward to verify that S'' is an independent set of G , otherwise an edge $\{v_i, v_j\}$ in $G - S''$ will cause a $K_{1,s}$ induced by $I_i \cup \{v_i, v_j\}$ in $G' - S''$, which is a contradiction. Now, the statements follow from Proposition 3.1, Lemma 3.2 and the fact that the reduction is both linear and linear parameterized.

Now, we give a reduction from $K_{1,s+1}$ -FREE VERTEX DELETION on triangle-free graphs to s -DIAMOND-FREE EDGE DELETION (s -DIAMOND-FREE EDGE EDITING) on K_4 -free graphs.

Reduction: Let $s \geq 1$ be any fixed integer. Let (G, k) be an instance of $K_{1,s+1}$ -FREE VERTEX DELETION such that G is triangle-free. Introduce a new vertex w and make it adjacent to all the vertices in G . Let the resultant graph be G' and let the reduced instance of DIAMOND-FREE EDGE DELETION (DIAMOND-FREE EDGE EDITING) be (G', k) .

Observation 3.4 For any fixed integer $s \geq 1$, let (G, k) be an instance of $K_{1,s+1}$ -FREE VERTEX DELETION on triangle-free graphs and let G' be obtained by the reduction described above.

- (i) G' is K_4 -free.
- (ii) Let $D \subseteq V(G')$. Then, D induces an s -diamond in G' if and only if $w \in D$ and $D \setminus \{w\}$ induces a $K_{1,s+1}$ in G .
- (iii) Let $S \subseteq V(G)$ and let F be the set of all edges $\{w, v_i\}$, where $v_i \in S$. Then, $G - S$ is $K_{1,s+1}$ -free if and only if $G' - F$ is s -diamond-free.

Proof. (i) Implied by the fact that G is triangle-free.

- (ii) Assume that D induces an s -diamond in G' . Since G is triangle-free, $w \in D$ and w must be a vertex with degree $s + 2$ in the s -diamond $G'[D]$. Hence $D \setminus \{w\}$ induces a $K_{1,s+1}$ in G . The other direction is straight-forward.
- (iii) Let $G - S$ be $K_{1,s+1}$ -free. By (ii), every induced s -diamond in G' is formed by an induced $K_{1,s+1}$ in G and w . Hence every s -diamond in G' is hit by F . Since G' is K_4 -free, no new s -diamond is created in G' by deleting edges. Hence $G' - F$ is s -diamond-free. Conversely, let $G' - F$ be s -diamond-free. By (ii), every induced $K_{1,s+1}$ in G with w forms an s -diamond in G' . Hence S hits all induced $K_{1,s+1}$ s in G .

Theorem 3.5 *For any fixed $s \geq 1$, s -DIAMOND-FREE EDGE DELETION and s -DIAMOND-FREE EDGE EDITING are NP-complete even on K_4 -free graphs. Further, these NP-complete problems can be solved neither in time $2^{o(|G|)}$ nor in time $2^{o(k)} \cdot |G|^{O(1)}$.*

Proof. We reduce from $K_{1,s+1}$ -FREE VERTEX DELETION on triangle-free graphs. Let (G, k) be an instance of $K_{1,s+1}$ -FREE VERTEX DELETION such that G is triangle-free. We apply the reduction described above to obtain an instance (G', k) of s -DIAMOND-FREE EDGE DELETION (s -DIAMOND-FREE EDGE EDITING). By Observation 3.4(i), G' is K_4 -free. We claim that (G, k) is a yes-instance of $K_{1,s+1}$ -FREE VERTEX DELETION if and only if (G', k) is a yes-instance of s -DIAMOND-FREE EDGE DELETION (s -DIAMOND-FREE EDGE EDITING).

Let $S \subseteq V(G)$ be such that $|S| \leq k$ and $G - S$ is $K_{1,s+1}$ -free. Let $F \subseteq E(G')$ be defined as the set of all edges $\{w, v_i\}$, where $v_i \in S$. Now, by Observation 3.4(iii), $G' - F$ is s -diamond-free.

Conversely, let (G', k) be a yes-instance of s -DIAMOND-FREE EDGE DELETION (s -DIAMOND-FREE EDGE EDITING). Let $T \subseteq E(G')$ ($T \subseteq [V(G')]^2$) be such that $|T| \leq k$ and $G' \triangle T$ is s -diamond-free. For every (potential) edge $\{v_i, v_j\}$ of G in T , replace $\{v_i, v_j\}$ with either $\{w, v_i\}$ or $\{w, v_j\}$ in T . Let the resultant set of edges be T' . Clearly, $|T'| \leq k$. By 3.4(ii), every s -diamond in G' contains w . Therefore, since T hits every s -diamond in G' , T' hits every s -diamond in G' . Since G' is K_4 -free, no new s -diamond is created by deleting edges from G' . Hence, $G' - T'$ is s -diamond-free. Let S be the set of all vertices v_i such that $\{w, v_i\}$ is in T' . Then, by 3.4(ii), $G - S$ is $K_{1,s+1}$ -free. Now, the statements follows from Proposition 3.3 and the observation that the reduction we give is both linear and linear parameterized.

We observe that, in the proof of Theorem 3.5, even for the editing problem, for every solution (the set of edges to be deleted and the set of edges to be added) of (G', k) , has a corresponding solution which contains only the set of edges to be deleted. Hence we get the following corollary.

Corollary 3.6 *For any fixed integers $s \geq 1$ and $t \geq 4$, $\{s\text{-DIAMOND}, K_t\}$ -FREE EDGE DELETION and $\{s\text{-DIAMOND}, K_t\}$ -FREE EDGE EDITING are NP-complete even on K_4 -free graphs. Further, these problems can be solved neither in time $2^{o(|G|)}$ nor in time $2^{o(k)} \cdot |G|^{O(1)}$.*

4 Concluding Remarks

Consider the graph G in Figure 3. A big circle denotes a clique of k vertices. There are k of them. Outside the large cliques there are only four vertices which induces a diamond. One of those vertices is adjacent to all the vertices in the large cliques (thick edge denotes this) and another vertex in the diamond is adjacent to exactly one vertex from each large clique. We observe that G has $k^2 + 4$ vertices. None of our rules reduces the size of this graph and a $O(k)$ of such structures in a graph causes $O(k^3)$ vertices. We believe that rules to tackle this structure is the key to obtain a smaller kernel for DIAMOND-FREE EDGE DELETION.

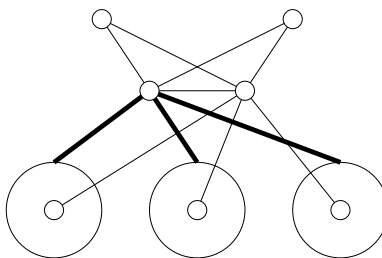


Fig. 3: A structure to dismantle for a smaller kernel for DIAMOND-FREE EDGE DELETION

Open Problem 1 *Does DIAMOND-FREE EDGE DELETION admit a kernel of $O(k^2)$ vertices?*

We have proved the hardness and lower bounds for s -DIAMOND-FREE EDGE DELETION, for any fixed $s \geq 1$. The vertex-split rule is safe to apply for these problems. Studying the structural properties of s -diamond-free graphs may help us to obtain a polynomial kernels for these problems when $s \geq 2$.

Open Problem 2 *Does s -DIAMOND-FREE EDGE DELETION admit a polynomial kernel when $s \geq 2$?*

Polynomial kernelization of CLAW-FREE EDGE DELETION is considered as a difficult problem [6, 9] in this area. One of the difficulties with this problem is that the characterization of claw-free graphs is quite complicated. A paw graph

is a graph obtained by adding an edge between two non-adjacent vertices in a claw. It is known that every component in a paw-free graph is either triangle-free or complete multipartite [18]. Can we use this to obtain a polynomial kernel for PAW-FREE EDGE DELETION?

Open Problem 3 *Does PAW-FREE EDGE DELETION admit a polynomial kernel?*

References

- [1] N. R. Aravind, R. B. Sandeep, and Naveen Sivadasan. Parameterized lower bounds and dichotomy results for the NP-completeness of H -free edge modification problems. In *LATIN 2016*.
- [2] N. R. Aravind, R. B. Sandeep, and Naveen Sivadasan. On polynomial kernelization of H -free edge deletion. In *Parameterized and Exact Computation - 9th International Symposium, IPEC*, pages 28–38, 2014.
- [3] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.
- [4] Hans L Bodlaender and Babette de Fluiter. On intervalizing k -colored graphs for DNA physical mapping. *Discrete Applied Mathematics*, 71(1):55–77, 1996.
- [5] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.
- [6] Leizhen Cai and Yufei Cai. Incompressibility of H -free edge modification problems. *Algorithmica*, 71(3):731–757, 2015.
- [7] Yufei Cai. Polynomial kernelisation of H -free edge modification problems. Mphil thesis, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China, 2012.
- [8] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [9] Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, Erik Jan van Leeuwen, and Marcin Wrochna. Polynomial kernelization for removing induced claws and diamonds. *WG*, 2015.
- [10] P. G. Drange, Markus Dregi, and R. B. Sandeep. Compressing bounded degree graphs. In *LATIN 2016*.
- [11] Pål Grønås Drange and Michał Pilipczuk. A polynomial kernel for trivially perfect editing. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Proceedings*, pages 424–436, 2015.
- [12] Ehab S El-Mallah and Charles J Colbourn. The complexity of some edge deletion problems. *Circuits and Systems, IEEE Transactions on*, 35(3):354–362, 1988.
- [13] Michael R. Fellows, Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. Graph-based data clustering with overlaps. *Discrete Optimization*, 8(1):2–17, 2011.
- [14] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- [15] Christian Komusiewicz. Tight running time lower bounds for vertex deletion problems. *arXiv preprint arXiv:1511.05449*, 2015.
- [16] Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. *Discrete Optimization*, 10(3):193–199, 2013.

- [17] Bojan Mohar. Face covers and the genus problem for apex graphs. *Journal of Combinatorial Theory, Series B*, 82(1):102–117, 2001.
- [18] Stephan Olariu. Paw-fee graphs. *Inf. Process. Lett.*, 28(1):53–54, 1988.
- [19] Svatopluk Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15(2):307–309, 1974.
- [20] Mihalis Yannakakis. Edge-deletion problems. *SIAM J. Comput.*, 10(2):297–309, 1981.